SUCCESSFUL PREDICTION OF HORSE RACING RESULTS USING A NEURAL NETWORK

N M Allinson and D Merritt

1 Introduction

This contribution has two main sections. The first discusses some aspects of multi-layer perceptrons, while the second outlines an application - namely the prediction of horse racing results.

2 Multi-layer perceptrons

Most application work within neural computing continues to employ multi-layer perceptrons (MLP). Though many variations of the fully interconnected feed-forward MLP, and even more variations of the back propagation learning rule, exist; the first section of this paper attempts to highlight several properties of these standard networks.

2.1 Adaptive Mappers

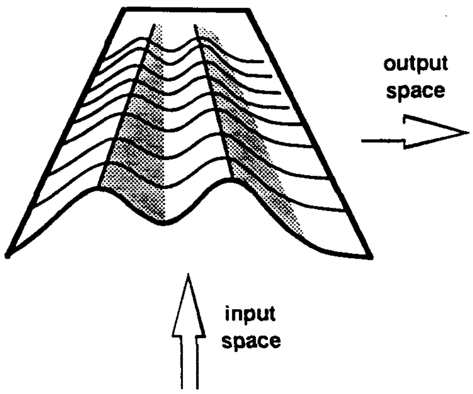


Fig. 1

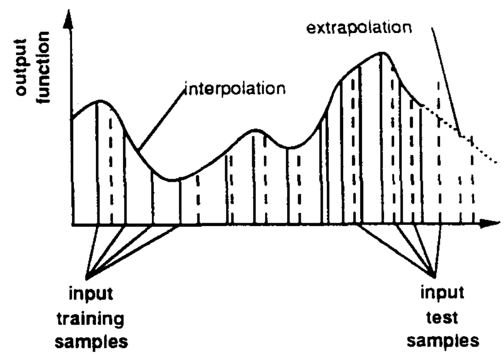


Fig. 2

MLPs act as adaptive mappers (Fig. 1) that learn, through supplied examples, to map from one function in the input space to another in the output space. This ability can be exploited for supervised pattern classification. Note that the learning must be supervised, in that input pattern vectors and output target vectors need to be presented in pairs. Such supervised learning implies that we already possess an *a priori* model of the underlying processes involved. Fig. 2 illustrates this mechanism for the one-dimensional situation. The MLP interpolates a smooth function between the supplied input training examples. Note that the Nyquist's sampling theory has a part to play, in that training samples must be sufficiently close to each other and well distributed. Therefore the MLP acts as an interpolator and, possibly, a poor extrapolator. The use of sigmoid compressive functions in the processing elements or neurons of the network help to ensure a smooth interpolation curve. For many applications this is ideal. Most physical phenomena are convoluted by Gaussian shaped corrupting functions. But,

Department of Electronics, University of York, York, YO1 5DD

the application of sigmoid compressive functions to training data, which we may term *logical* (see Fig. 3), is much less likely to be successful.

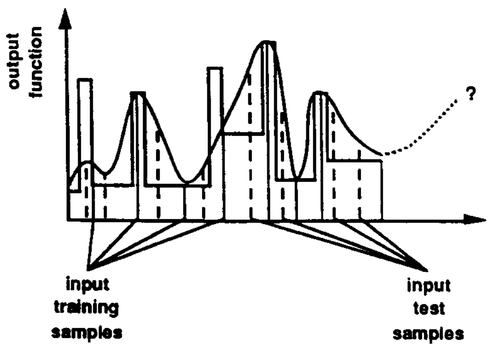


Fig. 3

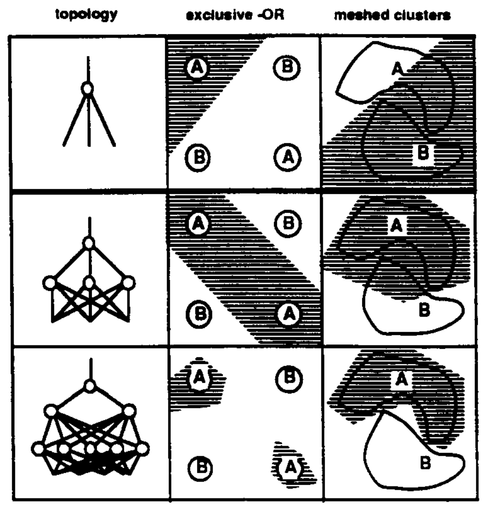


Fig 4

2.2 Network complexity

The abilities of MLPs in isolating regions of pattern space as a function of the number of layers are well known. Fig. 4 illustrates the general form of the discriminant functions possible. These are for the case of a hard non-linearity function in the neurons where, in the n-dimensional situation, hyper-planes can be constructed. For a single layer network, only a simple partitioning of the space is possible; for two layers, convex open or closed region of the space; and for three layers, any arbitrary shaped region or regions. The ability to cope with *hidden structures* (ie, the OR-ing of disjoint clusters) is one of the most significant abilities of MLPs. This is only possible in a supervised learning system. The use of smooth non-linearities that permit the application of the generalised delta rule (ie, differentiable) cause the region boundaries to become fuzzy (Fig. 5), which has implications for the generalisation abilities of the resultant MLP. For example, a test pattern can contribute to two or more regions. Extensions to a greater number of layers may help in successful learning since we are expecting a reduced complexity of non-linear mapping in each layer.

A question often discussed concerns the number of hidden units (ie, the number of neurons in layers which are not input or output layers). In general, there is no way of predicting this number, as it depends on the detailed shape of the pattern clusters in the appropriate space. The number of units is not, as is often suggested, related to the inherent dimensionality of the input pattern vectors. The situation is illustrated in Fig. 6 for two cases of similar two-dimensional pattern clusters. It should be noted that the hyperplane discriminant boundaries extend throughout the pattern space. This may be a contributory factor to the difficulties in scaling MLPs to more complex problems. If
there are no means to decide the number of hidden units, what effect does changing the number of units have? Too few and the system will be unable to classify

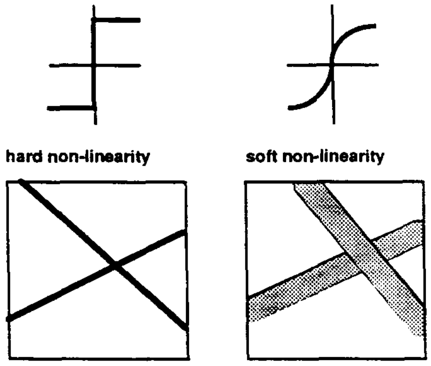


Fig. 5

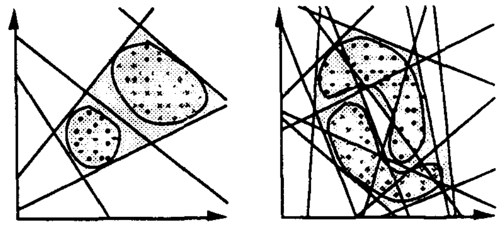


Fig 6

specific clusters (Fig. 7a), that is it will over generalise; too many and the system will be unable to generalise at all (Fig. 7c). Only you, as the designer, will be able to recognise satisfactory performance (Fig. 7b).

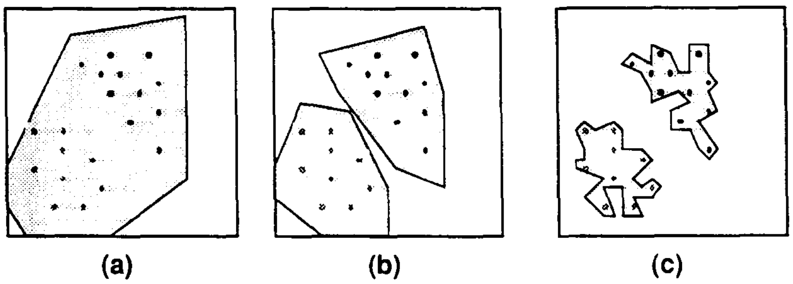
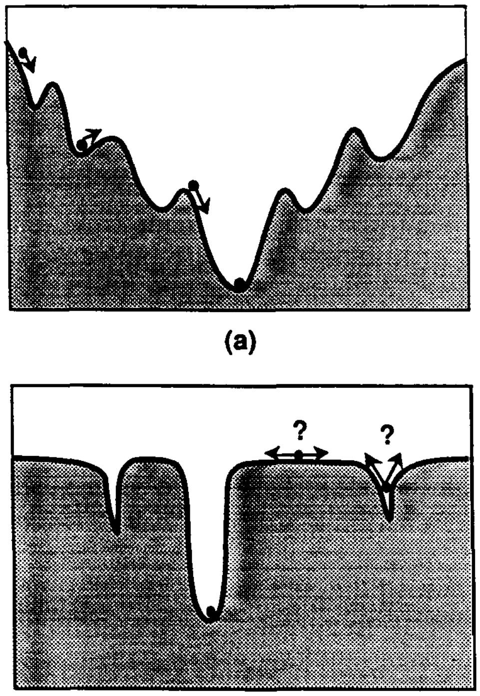


Fig 7

2.3 Learning

Learning is slow; usually many thousand cycles of training are required. It is difficult to move correctly discriminant functions (ie, hyperplanes) in space. In pattern clustering techniques, both traditional classification and neural network methods, single points are moved in space. This is much easier and, in part, explains the somewhat short learning times of such methods. The error surface (ie, the difference between the current output and target vectors, for each layer) can be a complex high dimensional function. As learning is essentially a method of gradient descent optimisation, then there is a high probability of becoming captured in a local minimum of this function. Varying the adaptation step size, use of momentum terms or higher order derivatives or use of added noise can help in overcoming this difficulty. A typical cross-section of the error surface is usually portrayed as in Fig. 8a; however it is more likely to take on the form of Fig. 8b (especially for the *logical* data as mentioned in 2.1). It is easy to choose a direction to go in order to find sea-level if you are stood on the top of a mountain. Much more difficult if you are stood on a horizontal plain.



(a)

(b)

Fig 8

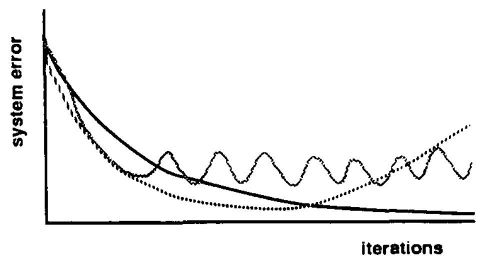


Fig 9

There is much evidence to suggest that MLPs learning algorithms are chaotic; certainly they are ill-conditioned. The same training schedule applied to the same network using the same data can lead to very different results. Not all attempts at learning results in the graceful decrease in error function with training time (Fig. 9). There are many reasons for unsuccessful learning - incorrect learning parameters, incorrect network complexity, over-learning (ie, over specialisation), inconsistent training data. It is difficult to diagnosis the reasons for failure, but experience has often highlighted problems in the training and application data sets. Variations on training include pre-processing the input data, prior to applying it to the network. There are considerable advantages in transforming the raw data into a set of basis functions, as we now require less complex non-linear mapping from the MLP.

2.4 Concluding remarks

The previous sections may seem over critical of MLPs but they are capable of providing a very useful pattern classification scheme for many situations - out performing traditional techniques. However, they are not a panacea for all problems. Perhaps their performance can be summarized as:

MLPs are good but not that good

3 Horse Racing Prediction

3.1 Introduction

From an initial position of ignorance, it was assumed that the skills employed by professional punters were much simpler than those used in the prediction of financial markets. Not so! It is essential for everyone involved in horse racing that valid information is available at the appropriate time to permit races to be run, bookmakers to operate and punters to place bets. Information commonly used include previous horse placings, weight and age of horse, names of jockey and trainer, length, going and value of each race. More specialised information is also available such as the distance a horse has travelled to get to the race[1]. Current information is provided not only in the national press but also in such publications as 'The Sporting Life.' Historical data is available, which consist not only of details of every race in the previous season but statistical surveys and rankings.

Predicting the outcome of individual races can range from choosing a horse by its name (ie, it has special meaning to the punter) to closely following the *form* and, of course, the all-important *insider tips*. Professional punters can make a living by betting, which suggests that prediction is possible. There is a long history of attempting to computerise prediction[2]. The best known of these is the Computer Straight Forecast, which is used by the betting shops to set their initial odds.

3.2 Problem domain

In order to increase the chances of success, only two-year old races on the Flat were considered. As most two-years will be sold at the end of the season and their performance will be reflected in their price, they tend to try to win in each of their outings (the same, cannot be said for older horses). But, a disadvantage is that at the beginning of the season these horses will have little or no form.

3.3 Network architecture

The learning data initially consisted of a pool of 200 horses - 100 winners and 100 losers. However, for all variations in the MLP architecture the prediction abilities were poor. The inclusion of 100 middle-placed horses in the training set overcame this problem. The number and nature of the input data is given in Table 1. The number of input units was varied. The effects of this for a typical unseen race are given in Table 2. The optimum number of training cycles was found to be about 500,000, and the non-linearities were conventional sigmoids. The final MLP architecture is shown in Fig. 10.

---

[1] Distance travelled is considered by the professional punter a very strong indicator of performance, but only if one horse is involved. For more than one horse, some may be there purely for company.

[2] It is rumoured that Charles Babbage attempted to predict horse races using his newly built analytical engine. But then he had problems funding his research as well.

Table 1
Input Data

Information	Description
Third last position	placing of horse in third last race
Second last position	placing of horse in second last race
First last position	placing of horse in last race
Age of horse	official age of horse in years. Horses' ages are incremented out of season
Weight	weight carried by horse, ie jockey and weights
Odds	opening odds on horse
Jockey percentage	percentage of wins for jockey over last three years
Jockey ranking	ranking of jockey compared with other jockeys at same course
Trainer percentage	percentage of wins for trainer over last three years
Trainer ranking	ranking of trainer compared with other jockeys at same course
Going	hardness of course (soft, good to soft, good, good to hard, hard)
Race value	value of total prize money for race

3.4 Results

The trained network was applied to many unseen races. Place bets were only laid if the network output prediction was greater than 0.9. Over a sample of ten races the typical returns were 30% to 50%, that is for a £10 stake, £13 to £15 was returned (no account of betting levy tax). As this work employed data from the previous season, all input information was available. For real-time application, then such information as the going will not be available until the day of the race but the odds available will almost certainly shortened from the opening odds.

3.5 Conclusion

This simple MLP application demonstrates the ability of such networks to make viable predictions, if care is taken over the choice of problem domain and 100% success is not

Table 2
MLP prediction for a typical race

Actual position of horse	Hidden units 5	12	20
1	0.483	0.957*	0.469
2	1.000*	1.000*	1.000*
3	0.492	0.961*	0.414
4	0.269	0.049	0.013
5	0.477	0.021	0.101
6	0.490	0.019	0.078
7	0.502	0.614	0.525
8	0.502	0.528	0.533
9	0.488	0.252	0.362
10	0.496	0.168	0.108
11	0.502	0.149	0.123

* - place bet

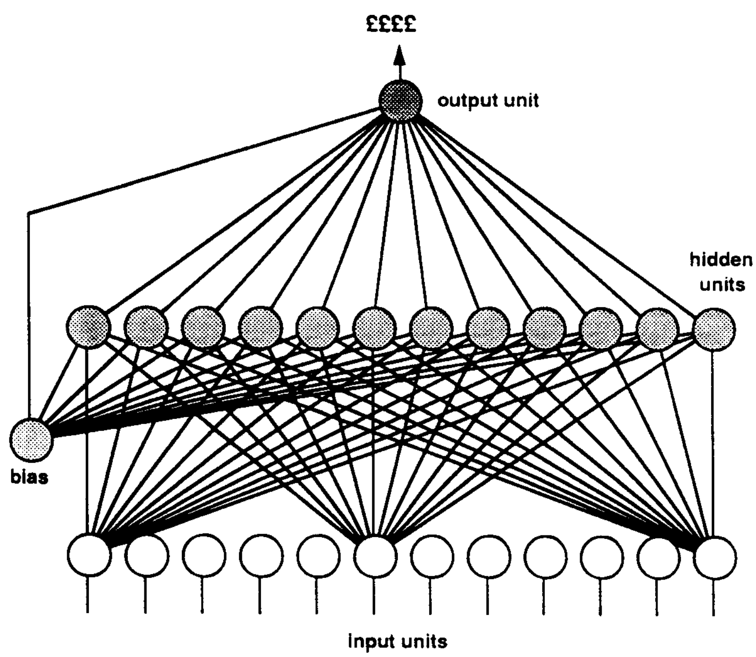


Fig. 10

required. The question most people now ask is "Why haven't you made your fortune?" Quite simply, the effort of entering data on several hundred horses, jockeys and trainers is not insignificant. This information does exist in computer readable form, but the owners seem reluctant to part with it!